

Modelos de Examen para Programacion II*

Mauricio Elian D. G.** Pedro Luis C. T.
*Facultad de Ingeniería en Ciencias de la Computación y
Telecomunicaciones - Universidad Autónoma Gabriel René Moreno*

Primer Parcial: Recursion

- 1) Cargar un vector con las palabras (vocales) de una cadena

Buenos dias a todos \rightarrow v[ueo,ia,a,oo]

Input:	Output:
Mucho Gusto Señor Programacion dos	v[uo,uo,eo] v[oaiao,o]

- 2) Cargar un vector de cadenas y devolver una cadena con el ultimo caracter de cada porción del vector.

v[hola|par|Tecno|x] \rightarrow s = arox

Input:	Output:
v[Amorio Carnaval Flojera Paz] v[Hola Tarot Tormenta Mar Cisne]	olaz atare

- 3) Cargar un vector con las palabras de una cadena donde la primera letra de cada palabra este en Mayuscula.

buenos dias a todos \rightarrow v[Buenos,Dias,A,Todos]

Input:	Output:
como estas xy ab podemos hablar en ingles	v[Como,Estas,Xy,Ab] v[Podemos,Hablar,En,Ingles]

- 4) Eliminar la ultima y la primera letra de cada palabra.

Buenos dias a hp todos \rightarrow ueno ia a odo

Input:	Output:
como estas xy ab podemos a en ingles	om sta odemo a ngle

*Estos modelos son una recopilación de exámenes de semestres pasados realizados en la cátedra del Ing. Contreras

**Para cualquier cambio, observación y/o sugerencia pueden enviarme un mensaje al siguiente correo: elianklk@gmail.com

5) Escribir la función:

$$Literal(n) = String$$

La cual devuelve una cadena de acuerdo a **n** que es un numero.

$$Litera(12) = 122$$

$$Litera(5) = 55555$$

Input:	Output:
341	33344441
27	227777777

6) Encontrar la palabra mas larga de una cadena.

$$x=hola mundo xyzw \rightarrow pal=mundo$$

Input:	Output:
podria estar comiendo sopa Eli sabe dos idiomas	comiendo idiomas

7) Cargar una matriz de la siguiente forma:

$$f(A, \text{fila}, \text{columna}, \text{razón})$$

f(5,0,0,1):

1	1	1	1	1
1	2	2	2	1
1	2	3	2	1
1	2	2	2	1
1	1	1	1	1

f(5,2,1,4):

0	0	0	0	0
0	0	0	0	0
0	4	4	4	4
0	4	5	5	4
0	4	4	4	4

f(4,0,1,1):

0	1	1	1
0	1	2	1
0	1	2	1
0	1	1	1

f(4,0,0,3):

3	3	3	3
3	4	4	3
3	4	4	3
3	3	3	3

8) Cargar una matriz de la siguiente forma:

$$f(A, \text{fila}, \text{columna}, \text{razón})$$

f(5,0,0,1):

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

f(5,0,0,4):

4	5	6	7	8
19	20	21	22	9
18	27	28	23	10
17	26	25	24	11
16	15	14	13	12

f(3,0,0,1):

1	2	3
8	9	4
7	6	5

f(4,0,0,3):

3	4	5	6
14	15	16	7
13	18	17	8
12	11	10	9

Segundo Parcial I

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(int v[], word n){
    word a,b,c;
    a=n; b=1; c=0;
    while(a>b){
        a=(a+b)/2;
        c=c+1;
    }
}
```
2.

```
void Pregu2(Cardinal n){
    if(n>2){
        Pregu3(n);
        Pregu2(n-1);
    }
}
```

$T(\text{Pregu3}(m)) = 3m + 4$ pc // 10mc
3.

```
void Pregu4(int v[], word n){
    int a,b,c,d,e;
    a=n; c=n;
    while(a>0){
        b=0;
        while(b<c){
            if(v[b]%2==0){
                b=n;
            }
            else
                b=b+1;
        }
        a=a-1;
    }
}
```

Segundo Parcial II

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(word n){
    word a,b,c,tot;
    a=1; tot=1;
    while(a<=n){
        b=1;c=a*a;
        while(b<=c){
            if((b%c)==0){
                tot=tot+1;
            }
            b=b+1;
        }
        a=a+1;
    }
}
```
2.

```
void Pregu2(int v[], word n){
    a=0;b=0;c=1;
    while(b<n){
        c=c*2; a=a+1;
        if(a>=n){
            a=0;b=b+1;
        }
        v[a]=c;
    }
}
```

Segundo Parcial III

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(word n)
{word a,b,c,tot;
 tot=1;
  for(a=1;a<=n;a++){
    for(b=1;b<=a;b++){
      if((b%a)==0){
        for(c=1;c<=b;c++){
          tot=tot+1;
        }
      }
    }
  }
}
```

2.

```
void Pregu2(Cardinal n)
{ Cardinal a;
  while(n>3){
    a = n;
    n = n*2;
    Pregu3(n/2);
    Pregu2(a-1);
  }
}
T(Pregu3(m))=2m pc // 6 mc
```

Segundo Parcial IV

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(int A[], word n)
{int i,j,k,con,z;
 k=1; z=0;
  for(i=1;i<=n;i++){
    con=0;
    k= A[i]*k;
    j=1;
    while(j>=1){
      con= con+1;
      j= j/2;
    }
    if(con>k)
      z= z+1;
  }
}
```

2.

```
void Pregu2(Word n)
{int x,cal,sum;
 x=n; i=0; sum=0;
  while(i<n){
    cal=ProxDup(i);
    sum=sum+cal;
    i=i+1;
  }
  sum=sum+ProxDup(n);
}
```

```
int ProxDup(int i)
{int j;
 j=i;
  while(j>0){
    j=j*2;
    i=i-2;
  }
  return j;
}
```

Segundo Parcial V

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Orde(int v[] , word n)
{word a,b;
 a= n; b= 1;
 while(a>b){
  Burbu(v,a);
  a= a-1;
 }
}

void Burbu(int v[] , int n)
{if(n>1){
 Burbu(n-1);
 if(v[n-1]>v[n-2]){
  int x = v[n-1];
  v[n-1] = v[n-2];
  v[n-2] = x;
 }
 }
}
```
2.

```
String Inver(String x)
{int n; String z;
 n= x.Length();
 if(n<2)
  z= x;
 else{
  String y = x.Substring(1,n/2);
  x.Delete(1,n/2);
  y= Inver(y);
  x= Inver(x);
  z= x+y;
 }
return z;
}
```
3.

```
Void OtroInv(int v[] , Cardinal n)
{int a,b;
 a= 0; b= n-1;
 while(a<b){
  int x= v[a];
  v[a]= v[b];
  v[b]= x;
  a= a+1; b=b-1;
 }
}
```

Segundo Parcial VI

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(int v[] , word n)
{word a,b,c;
 a= 0; b= 0; c=1;
 while(b<n){
  c= c*2; a= a+1;
  if(a>=n){
   a= 0; b= b+1;
  }
  v[a]= c;
 }
}
```
2.

```
void Pregu2(int v[] , Word n)
{int a,b,c;
 a= n; c= 0;
 while(a>0){
  a= a-1; b=n;
  while(b>c){
   if(v[b]%2==1)
    c= c+1;
   else
    b= b-1;
  }
 }
}
```
3.

```
void Pregu3(Cardinal n)
{if(n>1){
 Pregu4(n);
 Pregu3(n-1);
 }
}

T(Pregu4(m))= 3m pc//5 mc
```

Segundo Parcial VII

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(int v[] , word n)
{word a,b,x;
 a= 1; b= n;
 while(a<=b){
   if(v[b]>v[a]){
     x= v[a];
     v[a]= v[b];
     v[b]= x;
   }
   a++;
   b--;
 } }
```
2. Escribir un proceso para ordenar los elementos de inserción. Luego calcular su tiempo de ejecución realizando su respectiva tabla de conteo.
3.

```
void Pregu3(int v[] , word n){
 int a,b,c,x;
 a= 1; b= n;
 while(a<=b){
   c= (a+b)/2;
   if(P(v[c])==true){
     a= c+1;
   } else{
     b= b-1;
   }
 } } T(P(x))=2;
```

Segundo Parcial VIII

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
void Pregu1(int v[] , word n)
{word a,b,c;
 a= 1; b= n; c= 1;
 while(c<n){
   c= c*2;
   a= a+1;
   v[a]= c;
 } }
```
2.

```
void Pregu2(int v[] , word n)
{int a,b,c;
 a= n; c=0;
 while(a>0){
   a= a-1; b= a;
   while(b>c){
     if(v[b]%2 == 0)
       c= c+1;
     else
       b= b-1;
   } } }
```
3.

```
void Pregu3(Cardinal n)
{ if(n>2){
   Pregu3(n-1);
   Pregu4(n);
 } }
T(Pregu4(m))=2m pc // 6 mc
```

Segundo Parcial IX

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1.

```
int Pregunta1(int N)
{int A,B,i,j,k;
  A= 0; i= 1; k= 1;
  while(i<=N){
    B= i;
    j= 1;
    while(j<= B){
      B= B*k;
      j++;
    }
    A= A+B;
    i++;
  }
  return A;
}
```

2.

```
void Pregu2(int v[] ,
  Cardinal n)
{ if(n>2){
  Pregunta2(v,n-1);
  n= n-1;
  Pregunta2(v,n);
}
}
```

3.

```
void Pregunta3(int v[] ,
  Cardinal n)
{ Cardinal i,j,k;
  k=1;
  while(k<n){
    i=1; j=n;
    while(i<=j){
      if(v[i]<v[j])
        i= i*2;
      else
        j= j/2;
    }
    k= k+1;
  }
}
```

Segundo Parcial X

Calcular el tiempo de ejecución de los siguientes algoritmos realizando sus respectivas tablas de conteo

1. Escribir un algoritmo para calcular

la diferencia entre el mayor y menor
de los datos de un vector

2.

```
void Pregu2(int A[] , Word n)
{int a,b,k,m,z;
  a = 0;b = n-1; m = 0;
  while(a<b){
    k = n;
    while(k>0){
      k = k-1;
      m = m+1;
    }
    a = (b+a)/2 + 1;
    b = b-k;
  }
}
```

3.

```
void Pregunta3(Word n)
{int x,z;
  x = 0; z = n;
  while(x<n){
    Pro(z);
    x = x+1;
    Pro(x);
  }
}
T(Pro(m)) = 3m+4;
```